

file AF

TRANSMITTAL OF APPEAL BRIEF (Large Entity)

Docket No.  
NCR.0027US

In Re Application Of: Gregory H. Milby et al.

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
09/733,530	12/08/00	Samuel G. Rimmel	21906	2175	

Invention: Managing Allocation Of Temporary and Permanent Files In A Database System

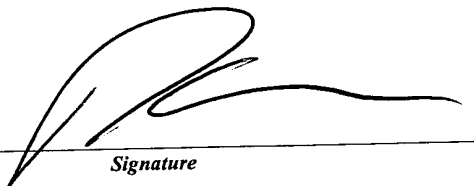
COMMISSIONER FOR PATENTS:

Transmitted herewith in triplicate is the Appeal Brief in this application, with respect to the Notice of Appeal filed on

The fee for filing this Appeal Brief is: \$340.00

- ☐ A check in the amount of the fee is enclosed.
- ☒ The Director has already been authorized to charge fees in this application to a Deposit Account.
- ☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. 50-1673 (9362)
- ☐ Payment by credit card. Form PTO-2038 is attached.

**WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.**

  
Signature

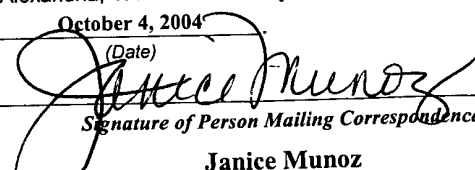
Dated: October 4, 2004

Dan C. Hu, Reg. No. 40,025  
TROP, PRUNER & HU, P.C.  
8554 Katy Freeway, Suite 100  
Houston, Texas 77024  
(713) 468-8880  
(713) 468-8883 (fax)

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on

October 4, 2004

(Date)

  
Signature of Person Mailing Correspondence

Janice Munoz

Typed or Printed Name of Person Mailing Correspondence

CC:



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants: Gregory H. Milby et al.

Serial No.: 09/733,530

Filed: December 8, 2000

For: MANAGING ALLOCATION OF  
TEMPORARY AND PERMANENT  
FILES IN A DATABASE SYSTEM

§ Group Art Unit: 2175  
§  
§  
§ Examiner: Samuel G. Rimmel  
§  
§  
§ Atty. Dkt. No.: 9362 (NCR.0027US)  
§  
§

Mail Stop Appeal Brief - Patents  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

APPEAL BRIEF PURSUANT TO 37 C.F.R. § 41.37

Sir:

The final rejection of claims 1-29 is hereby appealed.

**I. REAL PARTY IN INTEREST**

The real party in interest is the NCR Corporation by virtue of the assignment recorded at  
reel/frame 011373/0853.

**II. RELATED APPEALS AND INTERFERENCES**

None.

**III. STATUS OF THE CLAIMS**

Claims 1-29 have been finally rejected and are the subject of this appeal.

**IV. STATUS OF AMENDMENTS**

No amendments have been submitted after final rejection.

10/08/2004 MHEKONEN 00000007 501673 09733530  
01 FC:1402 340.00 DA

Date of Deposit: October 4, 2004  
I hereby certify under 37 CFR 1.8(a) that this correspondence is being  
deposited with the United States Postal Service as **first class mail** with  
sufficient postage on the date indicated above and is addressed to the  
Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313.  
*Janice Munoz*  
Janice Munoz

## V. SUMMARY OF THE INVENTION

Independent claim 1 recites a database system having a persistent data storage device storing a first file management context and having a pool of storage elements, and a non-persistent memory storing a second file management context. The first file management context is to indicate allocated permanent files in the pool of storage elements, and the second file management context is to indicate allocated temporary files and permanent files in the pool of storage elements.

Independent claim 16 recites a method for use in a database system having a persistent storage device and a non-persistent memory, where the method comprises storing a first file management context in the persistent storage device, storing a second file management context in the non-persistent memory, and updating both the first and second file management contexts to allocate a permanent file. The second file management context is updated without updating the first file management context to allocate a temporary file.

Independent claim 23 recites an article comprising at least one storage medium containing instructions that when executed cause a system to store a first file management context to indicate allocation of temporary and permanent files, and to store a second file management context to indicate allocation of permanent files.

As described by the specification, a permanent storage region and a temporary storage region are part of the same pool of storage resources in each storage unit. As one example, a pool of storage resources includes blocks or pages (generally referred to as “storage elements”) in the storage unit. Specification, p. 4, lines 9-12.

File management contexts are employed such that a block or a group of blocks can be selectively allocated to either permanent or temporary storage. Two groups of file management contexts are maintained: a first group 38 (Figures 1, 2) (referred to as “persistent file

management context”) is stored in a persistent storage unit 30; and a second group 26 (Figures 1, 2) (referred to as “non-persistent file management context”) is stored in a non-persistent memory 52 of each node 16. Each storage unit 30 is a persistent storage unit that maintains stored data even if power is removed from the database system or if the database system is reset. Specification, p. 4, lines 13-23.

The persistent file management context 38 includes maps 106 and 108 to allocate storage blocks or pages to permanent files. The non-persistent file management context 26 also includes maps 110 and 112 to allocate storage blocks or pages to both permanent and temporary files. Specification, p. 5, line 24-p. 6, line 6. A permanent file is considered “permanent” in the sense that they are not lost in the event of system crashes, resets, or power cycles. However, permanent files can be changed or deleted by user requests during normal operation of the database system. Specification, p. 4, lines 1-8.

Even though the same pool of storage elements is used to store both temporary and permanent files, the control logic in a data server of the database system performs different management functions depending on whether a flag that is set in a map corresponds to a temporary file or a permanent file. For a permanent file, the control logic in the data server performs transaction locking, logging of data changes made to the maps, and performs tasks to provide data consistency across restart boundaries. However, if a temporary file is allocated, and non-persistent map 110 and 112 are updated without updating the persistent maps 106 and 108, the control logic in the data server does not perform transaction logging, logging of data changes made to the maps, and performing tasks to provide data consistency across restart boundaries. Specification, p. 6, lines 7-18.

Storage capacity is made more efficient by combining the pool of storage elements to store both temporary and permanent files. At the same time, management overhead is reduced by performing certain tasks only if permanent files are allocated. Specification, p. 6, lines 19-21.

## **VI. ISSUE**

- A. Claims 1-29 stand rejected under 35 U.S.C. § 102(e) over Zheng.**

## **VII. ARGUMENT**

- A. Claims 1-29 stand rejected under 35 U.S.C. § 102(e) over Zheng.**

There are at least two reasons why the present claims are allowable over Zheng. First, Zheng does not qualify as prior art, as the Declaration under C.F.R. 1.131 (hereinafter “Rule 131 Declaration”), submitted on January 7, 2004, has established an invention date of the present invention that is prior to the effective § 102(e) date of Zheng. Moreover, even if Zheng qualifies as prior art, at least claims 1-22 and 24-29 are not anticipated by Zheng.

### **RULE 131 DECLARATION**

The Rule 131 Declaration established an invention date prior to September 26, 2000, which is the § 102(e) date of Zheng. The Examiner considered the Rule 131 Declaration as being “ineffective in overcoming” Zheng as a prior art reference. Appellant respectfully submits that the Examiner has committed legal error in objecting to the Rule 131 Declaration. The first point of error made by the Examiner is the assertion of inadequacy of the invention disclosure record attached as Exhibit A to the Rule 131 Declaration. The Examiner objected to the invention disclosure record because it was unsigned and undated, with signature and date blocks not filled in by the inventors or other witnesses. 3/30/2004 Office Action at 6. Appellant respectfully submits that nowhere in the Patent Office rules or M.P.E.P. is there any requirement that an invention disclosure record has to be signed by inventors. The Rule 131 Declaration,

signed by the inventors, expressly stated that the document attached as Exhibit A is a copy of the invention disclosure submitted by the inventors to NCR Corporation, the assignee of the present application. This sworn statement is sufficient to authenticate the invention disclosure record.

Furthermore, the M.P.E.P. itself allows dates of an exhibit to be removed or blocked off. *See* M.P.E.P. § 715.07 (8<sup>th</sup> ed., Rev. 1) at 700-231. “[I]f the applicant or patent owner does not desire to disclose his or her actual dates, he or she may allege that the acts referred to occurred prior to a specified date.” *Id.* The inventors in the Rule 131 Declaration have alleged that conception occurred before September 26, 2000, which is the § 102(e) date of Zheng. Thus, sufficient evidence has been provided to establish a conception date occurring before September 26, 2000.

The Examiner also committed error in objecting to the Rule 131 Declaration on the basis that no evidence regarding diligence was provided between the alleged date of conception (March 26, 2000) and the first contact with Appellant’s representative (Dan C. Hu) on August 23, 2000. Evidence of diligence between March 26, 2000 and August 23, 2000, is not required in the present case to overcome Zheng. “Under 37 CFR § 1.131, the critical period in which diligence must be shown begins just prior to the effective date of the reference or activity and ends with the date of a reduction to practice, either actual or constructive (*i.e.*, filing a United States patent application).” M.P.E.P. § 715.07(a) at 700-233. Therefore, the critical period in the present case is the time just prior to September 26, 2000 (the filing date of Zheng) and the filing of the patent application. The Examiner has not challenged the evidence regarding diligence between August 23, 2000 and the filing date of December 8, 2000. Therefore, it is respectfully submitted that the Rule 131 Declaration is adequate in overcoming Zheng as a prior art reference.

Because Zheng has been removed as a prior art reference, reversal of the final rejection of claims 1-29 over Zheng is respectfully requested.

#### CLAIM 23

Because of the disqualification of Zheng as prior art, withdrawal of the final rejection of claim 23 is respectfully requested.

#### CLAIMS 1-13, 15-21, 27, 28

Moreover, Zheng does not disclose the subject matter of claim 1, which recites a database system having:

- a *persistent* data storage device storing a first file management context and having a pool of storage elements (with the first file management context indicating allocated permanent files in the pool of storage elements); and
- a *non-persistent* memory storing a second file management context (with the second file management context to indicate allocated temporary files and permanent files in the pool of storage elements).

Figure 3 of Zheng shows a block index (labeled 39 in Figure 1 of Zheng). The Examiner indicated that the first 3 columns of the block index shown in Figure 3 of Zheng constitute the first file management context recited in claim 1, and all of the columns of the block index of Figure 3 of Zheng constitute the second file management context. 3/30/2004 Office Action at 6. The block index 39 is part of an in-memory file system index 37. Zheng, 6:32-33, 45-50. Because everything shown in Figure 3 is stored in memory, the structure of Figure 3 fails to satisfy the recitation in claim 1 that a *persistent* data storage device stores the first file management context, and a non-persistent memory stores a second file management context.



The Examiner noted that Figure 3 of Zheng illustrates a “relational table” and that the “relational table itself is a permanent fixture of the system.” 3/20/2004 Office Action at 6. The Examiner further stated that “[w]hile the bits within the table can be changed as needed, the table itself is never destroyed or erased.” *Id.* Consequently, the Examiner considered the relational table making up the block index to be the “persistent storage” of claim 1. Appellant respectfully disagrees with this assessment of the in-memory block index 39 of Zheng. Clearly, as shown in Figure 1 of Zheng, a distinction is made between data stored in memory and data stored on a disk 22. In-memory data, such as the block index 39 depicted in Figure 3, is intended to be lost when power to the system is shut off. Therefore, the statement by the Examiner that the “table itself is never destroyed or erased” does not accurately describe the block index 39, which is stored in memory and thus will be erased when system power is removed. The memory for storing the block index 39 therefore cannot be the “persistent data storage device” recited in claim 1.

Moreover, a table such as the block index table cannot be considered a storage device. A table is stored in a storage device. In the context of Zheng, the table making up the block index 39 is stored in memory, which is a volatile or non-persistent storage device. Therefore, the elements of claim 1 cannot be met by Zheng.

Similarly, with respect to independent claim 16, Zheng fails to disclose storing a first file management context in a *persistent* storage device, and storing a second file management context in a *non-persistent* memory. Also, Zheng fails to disclose updating both the first and second file management contexts to allocate a permanent file, and updating the second file management context without the first file management context to allocate a temporary file.

With respect to independent claim 27, Zheng does not disclose an article comprising at least one storage medium containing instructions that when executed cause a system to store a

first file management context in non-persistent memory to indicate allocation of temporary and permanent files, and store a second file management context in persistent storage to indicate allocation of permanent files.

Withdrawal of the final rejection of claims 1-13, 15-21, 27, and 28 is respectfully requested for this additional reason.

#### CLAIMS 14, 22, 26, 29

Claim 14 recites that an access module performs at least one of a transaction locking and database logging operation *when updating the first file management context*, and the access module does not perform the transaction locking and database logging operations *when updating the second file management context but not updating the first file management context*. The Examiner pointed to the teachings of Zheng regarding the transaction log 33 (Figure 1), which is used to store before images of write operations (*see* Figures 10-11 of Zheng). However, there is no teaching by Zheng of performing this transaction logging when the first file management context is updated, but *not* performing this transaction logging when the second file management context is updated but the first file management context is not updated.

For this additional reason, claim 14 is not anticipated by Zheng. Claims 22, 26, and 29 are similarly not anticipated by Zheng. Reversal of the final rejections of claims 14, 22, 26, and 29 is therefore respectfully requested for this additional reason.

#### CLAIMS 24, 25

Claim 24, which depends from claim 23, recites the receiving of a request containing a flag to indicate a permanent file or a temporary file, where *both* the first and second file

management contexts are updated if the flag indicates a permanent file, and where the second file management context is updated without updating the first file management context if the flag indicates a temporary file. There is no teaching in Zheng of updating one or both of the first and second file management contexts based on the state of a flag in a request.


Reversal of the final rejection of claims 24 and 25 is respectfully requested for this additional reason.

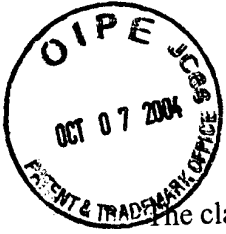
### VIII. CONCLUSION

In view of the foregoing, reversal of all final rejections and allowance of all pending claims is respectfully requested.

Respectfully submitted,

Date: 10-4-04

  
\_\_\_\_\_  
Dan C. Hu  
Registration No. 40,025  
TROP, PRUNER & HU, P.C.  
8554 Katy Freeway, Suite 100  
Houston, TX 77024  
Telephone: (713) 468-8880  
Facsimile: (713) 468-8883



## APPENDIX OF CLAIMS

The claims on appeal are:

1           1.       A database system comprising:  
2                   a persistent data storage device storing a first file management context and  
3       having a pool of storage elements; and  
4                   a non-persistent memory storing a second file management context,  
5                   the first file management context to indicate allocated permanent files in  
6       the pool of storage elements, and  
7                   the second file management context to indicate allocated temporary files  
8       and permanent files in the pool of storage elements.

1           2.       The database system of claim 1, wherein the first file management context  
2       is a subset of the second file management context.

1           3.       The database system of claim 1, further comprising a control module  
2       adapted to update an entry in the second file management context without updating an  
3       entry in the first file management context to allocate a temporary file.

1           4.       The database system of claim 3, wherein the control module is adapted to  
2       update an entry in both the first and second file management contexts to allocate a  
3       permanent file.

1           5.       The database system of claim 1, wherein the pool of storage elements  
2       comprises a pool of storage blocks.

1           6.       The database system of claim 5, further comprising a control module  
2       adapted to allocate one or more of the storage blocks to a temporary file or a permanent  
3       file.

1           7.     The database system of claim 5, wherein the first file management context  
2 contains a first storage identifier map and a first allocation unit map, the first storage  
3 identifier map indicating which storage identifiers have been allocated to permanent files,  
4 and the first allocation unit map indicating which storage blocks have been allocated to  
5 permanent files.

1           8.     The database system of claim 7, wherein the second file management  
2 context contains a second storage identifier map and a second allocation unit map, the  
3 second storage identifier map indicating which storage identifiers have been allocated to  
4 temporary and permanent files and the second allocation unit map indicating which  
5 storage blocks have been allocated to temporary and permanent files.

1           9.     The database system of claim 1, further comprising an access module  
2 containing the non-persistent memory.

1           10.    The database system of claim 9, wherein the access module comprises a  
2 data server to control access of the data storage device.

1           11.    The database system of claim 10, further comprising an application  
2 programming interface containing methods invocable by the data server to access the first  
3 and second file management contexts.

1           12.    The database system of claim 9, wherein the access module is adapted to  
2 copy the first file management context from the persistent data storage device to the non-  
3 persistent memory upon system restart.

1           13.    The database system of claim 9, further comprising:  
2                    one or more other access modules;  
3                    one or more other persistent storage devices accessible by the  
4   corresponding one or more other access modules; and  
5                    one or more other first and second file management contexts  
6   corresponding to the one or more other access modules.

1           14.    The database management system of claim 9, wherein the access module  
2   performs at least one of a transaction locking and database logging operation when  
3   updating the first file management context, and the access module is adapted not to  
4   perform the transaction locking and database logging operations when updating the  
5   second file management context but not updating the first file management context.

1           15.    The database management system of claim 1, wherein the permanent files  
2   contain user data and the temporary files contain results of queries.

1           16.    A method for use in a database system having a persistent storage device  
2   and a non-persistent memory, comprising:  
3                    storing a first file management context in the persistent storage device;  
4                    storing a second file management context in the non-persistent memory;  
5                    updating both the first and second file management contexts to allocate a  
6   permanent file; and  
7                    updating the second file management context without updating the first  
8   file management context to allocate a temporary file.

1           17.    The method of claim 16, further comprising maintaining the first file  
2   management context despite system reset, wherein the second file management context is  
3   lost due to the system reset.

1           18.     The method of claim 16, wherein the first file management context  
2 contains a storage identifier map to allocate storage identifiers and an allocation unit map  
3 to allocate blocks in the persistent storage device, and wherein updating the first file  
4 management context comprises updating the storage identifier map and the allocation  
5 unit map.

1           19.     The method of claim 18, wherein the second file management context  
2 contains a storage identifier map to allocate storage identifiers and an allocation unit map  
3 to allocate blocks in the persistent storage device, and wherein updating the second file  
4 management context comprises updating the storage identifier map and the allocation  
5 unit map.

1           20.     The method of claim 16, further comprising receiving a request, the  
2 request containing a flag to indicate allocation of a temporary file or a permanent file,  
3 wherein updating one or both of the first and second file management contexts is based  
4 on the flag.

1           21.     The method of claim 16, further comprising copying the first file  
2 management context to the non-persistent memory upon system startup.

1           22.     The method of claim 16, further comprising performing at least one of a  
2 transaction locking and database logging operation when updating the first file  
3 management context and not performing the transaction locking or database logging  
4 operation when updating the second file management context without updating the first  
5 file management context.

1           23.    An article comprising at least one storage medium containing instructions  
2   that when executed cause a system to:  
3                   store a first file management context to indicate allocation of temporary  
4   and permanent files; and  
5                   store a second file management context to indicate allocation of permanent  
6   files.

1           24.    The article of claim 23, wherein the instructions when executed cause the  
2   system to further:  
3                   receive a request containing a flag to indicate a permanent file or a  
4   temporary file;  
5                   update both the first and second file management contexts if the flag  
6   indicates a permanent file; and  
7                   update the second file management context without updating the first file  
8   management context if the flag indicates a temporary file.

1           25.    The article of claim 24, wherein the instructions when executed cause the  
2   system to update the first file management context by updating a first storage identifier  
3   map and a first allocation unit map, and update the second file management context by  
4   updating a second storage identifier map and a second allocation unit map.

1           26.    The database system of claim 1, further comprising a controller adapted  
2   to:  
3                   perform at least one of a transaction locking and database logging  
4   operation in response to detecting an update of the first file management context; and  
5                   not perform the transaction locking and database logging operations in  
6   response to detecting an update of the second file management context without an update  
7   of the first file management context.



1           27.    An article comprising at least one storage medium containing instructions  
2   that when executed cause a system to:  
3                   store a first file management context in non-persistent memory to indicate  
4   allocation of temporary and permanent files; and  
5                   store a second file management context in persistent storage to indicate  
6   allocation of permanent files.

1           28.    The article of claim 27, wherein the instructions when executed cause the  
2   system to:  
3                   update both the first and second file management contexts to allocate a  
4   permanent file,  
5                   update the first file management context without updating the second file  
6   management context to allocate a temporary file.

1           29.    The article of claim 28, wherein the instructions when executed cause the  
2   system to:  
3                   perform at least one of a transaction locking and database logging  
4   operation in response to detecting an update of the second file management context; and  
5                   not perform the transaction locking and database logging operations in  
6   response to detecting an update of the first file management context without an update of  
7   the second file management context.

## **EVIDENCE APPENDIX**

This appendix contains a copy of the Declaration Under 37 C.F.R. § 1.131 by Gregory H. Milby, Steven C. Grolemond, and Susan E. Choo, filed on January 7, 2004, with the Reply to Office Action Dated October 6, 2003. This Declaration was entered into the record by the Examiner on or before March 30, 2004 (the mailing date of the final Office Action).



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Gregory H. Milby, et al. § Group Art Unit: 2175  
Serial No. 09/733,530 § Examiner: Samuel G. Rimell  
Filed: December 8, 2000 §  
For: MANAGING ALLOCATION § Atty. Dkt. No.: 9362 (NCR.0027US)  
OF TEMPORARY AND §  
PERMANENT FILES IN A §  
DATABASE SYSTEM §

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**DECLARATION OF GREGORY H. MILBY,  
STEVEN C. GROLEMUND, AND SUSAN E. CHOO 37 C.F.R. 1.131**

Dear Sir:

I, Gregory H. Milby, Steven C. Grolemond, and Susan E. Choo, state as follows:

1. I am the inventor of the above-referenced application.
2. The document attached as Exhibit A is a copy of the invention disclosure I submitted to NCR Corporation, the assignee of the present application, regarding the invention described in the present application.
3. As set forth in the Declaration of Dan C. Hu (Exhibit B), the attorney who prepared the above-referenced application, NCR Corporation sent instructions to Dan C. Hu to prepare the application on or around August 23, 2000.
4. The above-referenced application was filed on December 8, 2000.
5. The attached documents establish that conception occurred before September 26, 2000, with constructive reduction to practice occurring on December 8, 2000. The attached documents also establish due diligence from conception to constructive reduction to practice.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: 1/5/2004

Gregory H. Milby  
Gregory H. Milby

Date: 1/3/2004

Steven C. Grolemond  
Steven C. Grolemond

Date: 12/30/2003

Susan E. Choo  
Susan E. Choo



## INVENTION DISCLOSURE RECORD

### PREPARATION & ROUTING INSTRUCTIONS

Complete and fill in every item. Write "none" or "unknown", if appropriate.

Use an additional blank page for any item where more space is needed.

Have your manager review and sign (items 9 and 10) before submitting to the NCR Law Department.

Submit original and one copy to: NCR Corporation, Intellectual Property Section, Law Department, ECD-2, 101 W. Schantz Avenue, Dayton, Ohio 45479. *Keep one copy for your file.*

#### LAW DEPARTMENT USE ONLY

Docket No.

9362

Date Received

Attorney

(1) Inventor(s)	Facility	Department	Phone Number
Gregory Milby	Rancho Bernardo	9515	858-485-3447
Steve Grolemond	Rancho Bernardo	9515	858-485-3425
Susan Choo	El Segundo	9515	310-524-7957

(2) Title of Invention (Preferably 10 words or less)

Shadow Map Management of DBS Temporary and Permanent Storage

(3) Product, Project Name or Class Number	(4) Date invention was First Conceived	(5) Actual or Anticipated Date of First Product Sale, Customer Availability, or Public Disclosure
TOR	3/26/2000	2001

(6) Description of the Invention

Please attach additional pages providing the following:

- Statement of problem solved by the invention** - Briefly state the problems your invention solves, its purposes and advantages, and how it differs from prior designs that you are aware of.
- Description of the invention** - Describe your invention in detail. Include and refer to sketches or diagrams and, if appropriate, attach documents such as previously prepared descriptions or specifications.
- Summary of Invention** - State what you regard at the present as the key inventive concept - i.e., the gist of your invention.

(7) Inventor Signature(s) (Each person listed in Item 1 above is an inventor and must sign and date.)			
Signature of Inventor	Date	Signature of Inventor	Date
Signature of Inventor	Date	Signature of Inventor	Date
(8) Witness Signatures (Two persons who are not inventors must read and understand this disclosure, and then sign and date.)			
Signature of Witness	Date	Signature of Witness	Date

#### FOR MANAGER USE ONLY

(9) Strategic Value of Patent Coverage (State what you regard as the strategic value to your business unit of having a patent for this invention - e.g., licensing revenue, preventing use by others, importance/breadth of the invention, etc.)			
(10) Reviewed and approved by			
Signature of Manager	Date	Manager Name (Please print)	Tentative Rating * (A, B, C, D, or U)

\* Ratings of "A" through "D" indicate relative value, with "A" being highest and "D" being lowest.  
A rating of "U" indicates the value is unknown.

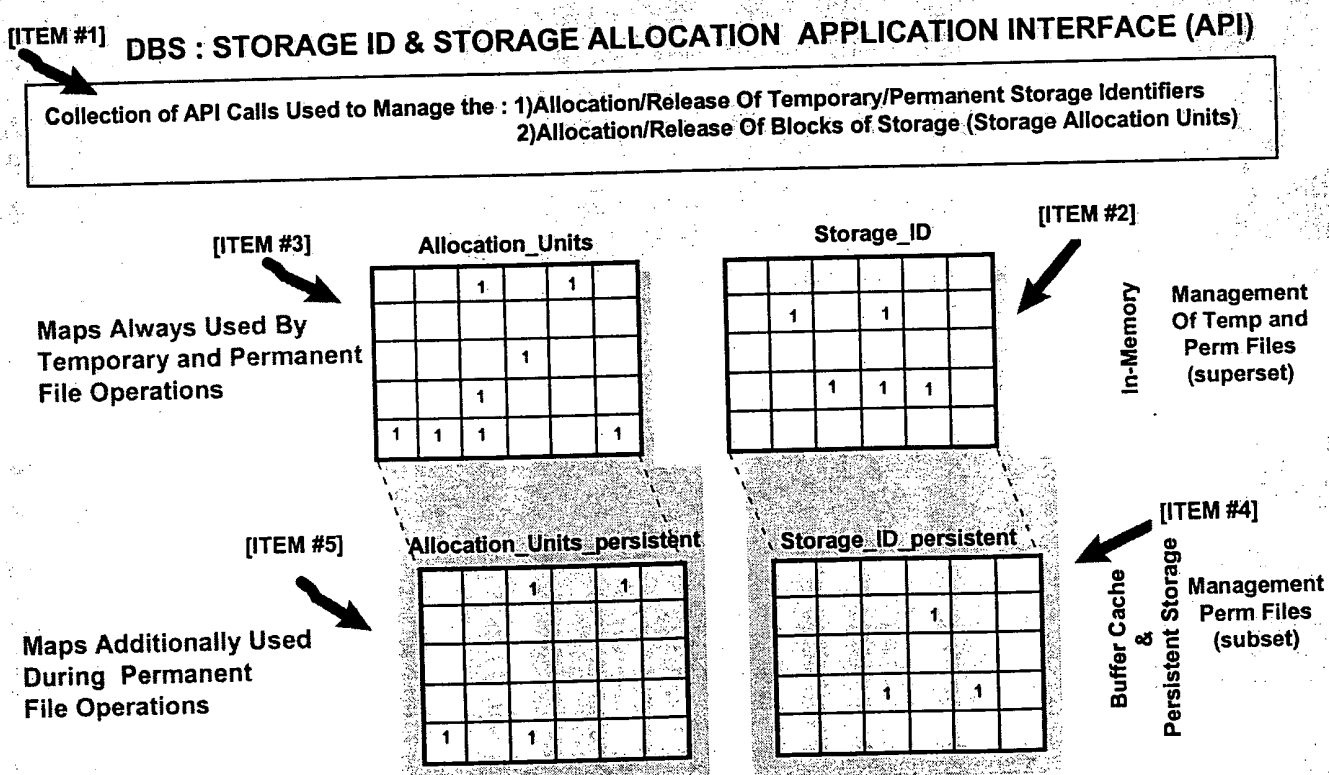
### Problem Solved By Invention

Modern database systems divide storage usage into system, permanent, and temporary data storage regions. Ideally, the same storage pool can be used to supply storage to satisfy all of these needs. Of the three, the two primary consumers of storage are temporary and permanent storage. User data is stored in Permanent storage, whereas Temporary Disk Storage houses the intermediate or final results of a client's query. Although both forms of storage contain, at times, user data, the behavior of the database logic, on behalf of the two types of storage, varies dramatically with respect to: 1) need to acquire Transaction Locks, 2) logging of data changes within the storage region, and, 3) levels of data consistency achieved in the event that a data recovery cycle is required. More specifically, it is expected that permanent file management context be consistent across system crashes or resets, whereas it is expected that temporary file management context be essentially erased in the event of a system crash or reset. A further motivation for segregating temporary and permanent file behavior is performance. A substantial performance gain can be achieved by allowing temporary file management operations to by-pass costly transaction lock, and database logging operations. The problem to be solved, and which is solved by this invention, is to enable both Permanent and Temporary Storage to be supplied by the same storage pool while at the same time providing an environment which enables the database software to treat the two storage types so dramatically different.

The invention provides three basic features: 1) providing the user with an application interface (API) and supportive logic that will enable them to manage both Permanent and Temporary storage from a common pool. 2) Providing logic to manage Temporary storage without requiring the acquisition of any transaction locks nor provoking any database logging activity, whereas the management of Permanent storage will be accompanied by a behavior which does result in both the acquisition of transaction locks, as well as provoking database logging activity. 3) Permanent File Management context will be consistent across system resets or crashes, whereas Temporary File Management context will basically be "erased".

### Description of Invention

## Temporary/Permanent Storage Shadow Maps



**Figure 1 : In-Memory and Persistent (shadow) maps used to manage Permanent/Temporary Storage**

The above diagram depicts the major components involved in the invention. In the diagram shown, a single application interface (API) is used to manage both permanent and temporary storage. This API provides management of both permanent (including system) and temporary storage which is being supplied from a common pool. The distinction between permanent and temporary space is accomplished by passing a flag

parameter, identifying the class of storage desired, with each storage management call. Amongst the services that the API must provide, are the service routines which:

1. Assign a unique identifier to identify each permanent or temporary storage file to be created.
2. Assign or Free blocks of storage (storage allocation units) to each of the temporary or permanent files.

These two services are implemented by creating two maps. One map keeps track of storage IDs which have already been allocated [ITEM #2] (and thus are in use), and another map keeps track of storage allocation units (contiguous series of blocks of storage) which have been allocated/assigned to the various permanent and temporary files [ITEM #3]. A common Application Interface (API) is used [ITEM #1] to access, search, and manipulate the maps. In the diagram, an allocated Storage Identifier, or allocated storage allocation unit, is depicted by flipping a flag bit to "1". The Storage ID [ITEM#2] and Allocation Unit [ITEM #3] maps are maintained solely within volatile main memory. The logic for the service routines ensures that writing to either the in-memory Storage ID or Allocation Unit map does not result in: 1) The acquisition of any Transaction Locks, 2) The writing of any database log records. The reason for this will soon become clear.

Each of the two principal maps are "shadowed": the storage ID map [ITEM #2] is shadowed by the corresponding persistent version of this map [ITEM #4]; the Allocation Unit map [ITEM #3] is shadowed by its corresponding persistent version [ITEM #5]. The shadow maps are persistent in that they are maintained in storage and read into memory, modified, then written back out to storage, whereas as the principal maps are maintained solely in memory. The logic which manipulates these maps ensures that any changes to the persistent versions of these maps are subject to: 1) Transaction Locking, 2) Logging of data changes made to the maps, and, 3) Guarantees of data consistency across restart boundaries. The reasons for this behavior will soon be made apparent.

At start-of-system, the persistent versions of the Storage ID [ITEM#4] and Allocation [ITEM#5] are read into memory. They are then copied in order to create the volatile in-memory versions of the Storage ID [ITEM#2] and Allocation [ITEM#3] maps. Thus, at start-of-system, each volatile in-memory map, and its corresponding persistent shadow, are equal.

Allocation of a storage identifier for a Temporary File is accomplished by first searching the in-memory Storage ID map [ITEM #2] for a free ID, and then setting a bit in the volatile in-memory Storage ID map [ITEM #2]. Allocation and assignment of units of storage to the new temporary file identified by storage ID is accomplished by first searching the in-memory Allocation map [ITEM#3] for free storage units, and setting the corresponding bits in the volatile in-memory Allocation Map [ITEM #3]. The setting of bits in the in-memory Storage ID [ITEM #2] and in-memory Allocation Map [ITEM #3] does not result in any transaction lock, or logging activity. Since these entries are made only in the volatile versions of the maps, the changes will be lost in the event of a system crash or reset. Upon completion of the reset cycle the persistent versions of the Storage ID [ITEM#4] and Allocation [ITEM#5] will be read into memory and then copied to form the volatile in-memory Storage ID [ITEM#2] and Allocation [ITEM#3] versions. This action has the effect of "returning" all of the temporary storage space that was in-usage prior to the reset, back to the system.

Allocation of a storage identifier for a Permanent File is accomplished by first searching the in-memory Storage ID map [ITEM #2] for a free ID, and then setting both a bit in the volatile in-memory Storage ID map [ITEM #2], as well as in the persistent Storage ID shadow map [ITEM #4]. Allocation and assignment of units of storage to the new permanent file identified by the storage ID is accomplished by first searching the in-memory Allocation map [ITEM#3] for free storage units, and then setting the corresponding bits in both the in-memory Allocation Map [ITEM #3] as well as the persistent Allocation shadow map [ITEM #5]. The logic that performs the writing to the persistent Storage ID [ITEM #4] and Allocation [ITEM #5] maps will also perform the necessary transaction lock and logging activity. Entries made in the persistent versions of the maps will also possess the property of being recoverable across a system crash or reset.

From the above discussion it can be deduced that the in-memory versions of the Storage ID [ITEM#2] and Allocation [ITEM#3] are a superset of their respective persistent versions, Storage ID [ITEM#4] and Allocation [ITEM#5]. It can be further deduced, that:

1. The in-memory maps represent the union of file management context for the temporary files and permanent files.
2. The persistent version of the maps contains the file management context for the permanent files.
3. The portion of the in-memory maps that are not contained in their corresponding persistent versions contains the file management context for the temporary files.

The requirement that the search for free IDs and free allocation units utilize the in-memory versions of the Storage ID [ITEM #2] and Allocation [ITEM #3] maps is what provides the user with the ability to allocate both permanent and temporary storage from a common pool.

### Summary of Invention

The invention "Shadow Map Management of DBS Temporary and Permanent Storage" provides the DBS designer with an approach for managing both Permanent and Temporary Storage in a seamless manner. The key to the design is the "shadowing" of the primary usage in-memory maps, Storage ID [ITEM #2] and Allocation [ITEM #3] maps, with their respective persistent counterparts, Storage ID [ITEM#4] and Allocation [ITEM #5].

With regards to the first claim of the invention: The in-memory versions of the maps are supersets of their corresponding persistent counterparts and contain the file management context for both permanent and temporary files. The persistent versions of the maps ("shadowed" versions) house only a copy of the permanent file management context. The claimed feature of providing the DBS designer with the ability to manage both temporary and permanent storage from the same pool, is being accomplished by the fact that the API logic utilizes exclusively the in-memory version of the maps when searching for free storage ID's and free storage allocation units, and the logic always sets bits in this in-memory version regardless of the type of storage being allocated.

With regards to the second claim of the invention: The behavior of the logic when flipping bits in the in-memory versions of the maps is orthogonal to the behavior which occurs when flipping bits in their persistent counterparts. Only the in-memory versions of the maps are used when assigning storage IDs and/or storage allocation units on behalf of a temporary file. The flipping of the bits within the in-memory maps results in no acquisition of any transaction locks and no database logging activity. The assignment of storage ID's and/or storage allocation units on behalf of a permanent file requires the flipping of bits within the in-memory versions of the maps as well as the flipping of the same bits within the persistent map counterparts. The flipping of bits within the persistent versions of the maps results in the acquisition of the appropriate transaction locks and in database logging activity. It is this difference in behavior, between flipping bits in the in-memory versions of the map and flipping bits in the persistent versions of the map, which leads to the invention claim that an environment has been provided which allows both permanent and temporary storage to be managed from a common pool, while also allowing the orthogonal behavior, with respect to whether or not transaction locking and database logging occurs while managing the two distinct storage classes.

The third invention claim, that the permanent file management context be consistent across database system crashes or resets, whereas the temporary file management context be vanquished upon a system crash or reset, is accomplished by housing the both the temporary and permanent file management context in the volatile in-memory versions of the maps, and housing a copy of only the permanent file management context with the persistent versions of the maps. Upon a system reset or system crash, the volatile versions of the maps disappear, leaving only the persistent versions of the maps. Thus the temporary file management context is simply, "erased".



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Gregory H. Milby, et al.	§	Group Art Unit:	2175
Serial No.	09/733,530	§	Examiner:	Samuel G. Rimell
Filed:	December 8, 2000	§		
For:	MANAGING ALLOCATION OF TEMPORARY AND PERMANENT FILES IN A DATABASE SYSTEM	§	Atty. Dkt. No.:	9362 (NCR.0027US)

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

DECLARATION OF DAN C. HU

Dear Sir:

I, Dan C. Hu, state as follows:

1. I am the patent attorney responsible for preparing the above-referenced patent application.
2. I received a disclosure (attached as Exhibit A to the Declaration of Gregory H. Milby, Steven C. Grolemond, and Susan E. Choo) for purposes of preparing a patent application on the above-referenced invention on or around August 23, 2000 (*see* Exhibit 1, attached).
3. Upon receiving the disclosure, it was immediately placed in line to be prepared.
4. Generally, the number of applications we had in preparation at the time required that it take about three months to prepare a draft of the application.
5. In this case, a first draft was completed by November 24, 2000.
6. The application was sent out for inventor review on November 24, 2000 (*see* Exhibit 2 attached).
7. Shortly thereafter, the inventor provided comments on the patent application, and the application was revised accordingly (*see* Exhibit 2 attached).
8. On December 6, 2000, a second draft of the application was sent to the inventors for their review (*see* Exhibit 3, attached).
9. The application was filed with the U.S. Patent Office on December 8, 2000.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date:

Jan. 7, 2004

  
\_\_\_\_\_  
Dan C. Hu

# **Monica L Jacobs**

---

**From:** Cowart, John D  
[jc151001@exchange.SanDiegoCA.NCR.COM]  
**Sent:** Wednesday, August 23, 2000 1:00 PM  
**To:** Dan Hu (TPH)  
**Subject:** New disclosure (NCR 9362)



9362.Milby.Grolemund.  
Choo.doc

\*\*\*\*\*CONFIDENTIAL/ATTORNEY-CLIENT PRIVILEGED\*\*\*\*\*

Dan,

Here is another disclosure that I would like you to handle, filing before the end of this year. The first two inventors (Milby, Grolemund) are in RB; the third is in ES. Please let me know if you want to take this one.

My review committee is supposed to meet next Friday (the 1st), and we'll be reviewing two more disclosures from Greg Milby. If we decide to file on those, I'd like to send them to you too.

Thanks.  
John

<<9362.Milby.Grolemund.Choo.doc>>

**Subject: 9362****Date:** Wed, 6 Dec 2000 11:27:42 -0500**From:** "Grolemund, Steven" <SG122251@exchange.SanDiegoCA.NCR.COM>**To:** hu@tphm.com

Dan,

I reviewed the attached document. I looks fine to me.

Here's the other information you asked for:

Full Name: Steven Charles Grolemund

Residence: Poway, California

Citizenship: United States

Steve

&gt; -----Original Message-----

&gt; From: Milby, Greg

&gt; Sent: Wednesday, November 29, 2000 4:28 PM

&gt; To: Grolemund, Steven; Choo, Susan

&gt; Subject: FW: 9362

&gt;

&gt;

&gt;

&gt; -----Original Message-----

&gt; From: Dan C. Hu [SMTP:hu@tphm.com]

&gt; Sent: Friday, November 24, 2000 11:59 AM

&gt; To: Milby, Greg

&gt; Cc: Cowart, John D

&gt; Subject: 9362

&gt;

&gt; Greg,

> Attached is a first draft of the above-referenced application for your  
> review. Drawings are being faxed to you.

&gt;

&gt; Please forward a copy to Steve Grolemund and Susan Choo.

&gt;

> Can you give me an estimated time to complete your review of this  
> application? Also, please let me know when you can complete your review  
> of 9470, which was sent to you a couple of weeks ago?

&gt;

&gt; Best Regards,

&gt;

&gt; Dan C. Hu

&gt;

&gt; TROP, PRUNER &amp; HU, P.C.

&gt; 8554 Katy Freeway, Suite 100

&gt; Houston, TX 77024

&gt;

&gt; (713) 468-8880 (phone)

&gt; (713) 468-8883 (fax)

&gt;

&gt; CONFIDENTIAL/ATTORNEY-CLIENT PRIVILEGED

&gt; ATTORNEY WORK PRODUCT

&gt; &lt;&lt;0027 Patent Application.doc&gt;&gt;

12/6/2000 10:34 AM

**TROP, PRUNER & HU, P.C.**  
INTELLECTUAL PROPERTY LAW ATTORNEYS

8554 Katy Freeway, Suite 100  
Houston, Texas 77024  
Bus: (713) 468-8880  
Fax: (713) 468-8883

# Fax

To: Susan Choo	From: Dan Hu
Co:	Pages: 6
Fax: 310/524-5515	Date: December 6, 2000
Re: 9362	

- ☐ Urgent    ☒ For Review    ☐ Please Comment  
☐ Confirmation copy will follow via First Class Mail  
☒ Confirmation copy will not follow

Message:

Attached are the figures for the above-referenced patent application.

● Notice: This information is intended to be for the use of the individual or entity named on this transmittal sheet. If you are not the intended recipient, be aware that any disclosure, copying, distribution, or use of the contents of this faxed information is prohibited. If you have received this facsimile in error, please notify the sender by telephone immediately so that arrangements can be made for the retrieval of the original document at no cost to you.

**Attorney Client Privileged - Attorney Work Product**

```

*****
*                                     P. 01 *
*                                     *
*          TRANSACTION REPORT          *
*                                     *
*          DEC-06-00 WED 04:15 PM      *
*                                     *
*   DATE  START  RECEIVER  TX TIME  PAGES TYPE  NOTE  M#  DP *
*-----*-----*-----*-----*-----*-----*-----*
* DEC-06 04:14 PM 18584852032 1'15"    6  SEND   OK    608 *
*-----*-----*-----*-----*-----*-----*-----*
*                                     *
*                                TOTAL :    1M 15S  PAGES:    6 *
*                                     *
*****

```

**TROP, PRUNER & HU, P.C.**

INTELLECTUAL PROPERTY LAW ATTORNEYS

8554 Katy Freeway, Suite 100  
Houston, Texas 77024  
Bus: (713) 468-8880  
Fax: (713) 468-8883

# Fax

<b>To:</b>	Greg Milby	<b>From:</b>	Dan C. Hu
<b>Company:</b>		<b>Date:</b>	December 6, 2000
<b>Fax:</b>	858/485-2032	<b>Pages:</b>	
<b>Your Re:</b>	9362	<b>Our Re:</b>	NCRC-0027-US

☐ Urgent   
 ☒ For Review   
 ☐ Please Comment   
 ☐ Please Reply   
 ☐ Confirm Receipt

**TROP, PRUNER & HU, P.C.**

INTELLECTUAL PROPERTY LAW ATTORNEYS

8554 Katy Freeway, Suite 100  
Houston, Texas 77024  
Bus: (713) 468-8880  
Fax: (713) 468-8883

# Fax

<b>To:</b>	Greg Milby	<b>From:</b>	Dan C. Hu
<b>Company:</b>		<b>Date:</b>	December 6, 2000
<b>Fax:</b>	858/485-2032	<b>Pages:</b>	
<b>Your Re:</b>	9362	<b>Our Re:</b>	NCRC-0027-US

☐ Urgent    ☒ For Review    ☐ Please Comment    ☐ Please Reply    ☐ Confirm Receipt

## MESSAGE:

Attached are the figures for the above-referenced patent application.

● **Notice:** This information is intended to be for the use of the individual or entity named on this transmittal sheet. If you are not the intended recipient, be aware that any disclosure, copying, distribution, or use of the contents of this faxed information is prohibited. If you have received this facsimile in error, please notify the sender by telephone immediately so that arrangements can be made for the retrieval of the original document at no cost to you.